202111718

Manual técnico

Walter Javier Santizo Mazariegos

walterjav19@gmail.com

Modelado de los Menús

```
from tkinter import *
from tkinter import ttk
import tkinter
from tkinter import messagebox
from tkinter.font import Font
```

Importamos la clase Tkinter que nos permite usar interfaces graficas y contenedores que facilitan la interacción del usuario

Se crearon 8 ventanas por medio de funciones las cuales contaron con la funcionalidad de regresar que guarda la ventana con la función withdraw y manda a llamar los componentes con deiconify

```
principal.deiconify()
ruta.withdraw()
```

Se usaron componentes como label, treeview, Entry, button etc..

Clase Curso

```
class Curso:
    def __init__(self,codigo,nombre,pre_requisito,obligatorio,semestre,creditos,estado):
        self.codigo=codigo
        self.nombre=nombre
        self.pre_requisito=pre_requisito
        self.obligatorio=obligatorio
        self.semestre=semestre
        self.creditos=creditos
        self.estado=estado
```

Iniciamos el constructor con los 7 atributos que deseamos inicializar 3 de tipo string y los 4 restantes de tipo int

```
lista_objetos=[]
codis=[]
```

Creamos 2 listas una lista almacenara a todos los cursos que van a ser de tipo objeto y una lista que contendrá string o los identificadores únicos de los cursos

métodos de la clase

Leer Archivo

```
def leer_archivo(path):
    archivo=open(path,"r",encoding="utf-8")
    lineas=archivo.readlines()
    if lista_objetos==[]:
        for i in range(len(lineas)):
            linea=lineas[i].split(",")
            for j in range(1):
                if linea[j] not in codis:
                    objeto=Curso(linea[j],linea[j+1],linea[j+2],linea[j+3],linea[j+4],linea[j+5],linea[j+6])
                    lista_objetos.append(objeto)
                    codis.append(linea[j])
                else:
                    indice=Curso.buscar_curso(linea[j])
                    lista_objetos[indice].nombre=linea[j+1]
                    lista_objetos[indice].pre_requisito=linea[j+2]
                    lista_objetos[indice].obligatorio=linea[j+3]
                    lista_objetos[indice].semestre=linea[j+4]
                    lista_objetos[indice].creditos=linea[j+5]
                    lista_objetos[indice].estado=linea[j+6]
```

Vamos a recibir como parámetro la ruta del archivo el cual se va a leer y será una variable llamada path posteriormente leeremos todo el archivo y lo almacenaremos en la variable líneas. Posteriormente se recorre cada línea en el archivo y ser hace un Split de cada línea para tener un arreglo que por cada posición se inicializara el constructor y posteriormente se agregara el objeto a una lista, si el objeto existe simplemente se actualiza el curso

```
else:
    for i in range(len(lineas)):
        linea=lineas[i].split(",")
        for j in range(1):
            if linea[j] in codis:
                indice=Curso.buscar_curso(linea[j])
                lista_objetos[indice].nombre=linea[j+1]
                lista_objetos[indice].pre_requisito=linea[j+2]
                lista_objetos[indice].obligatorio=linea[j+3]
                lista_objetos[indice].semestre=linea[j+4]
                lista_objetos[indice].creditos=linea[j+5]
                lista_objetos[indice].estado=linea[j+6]
            else:
                objeto=Curso(linea[j],linea[j+1],linea[j+2],linea[j+3],linea[j+4],linea[j+5],linea[j+6])
                lista_objetos.append(objeto)
                codis.append(linea[j])
```

Cuando ya existan varios cursos en la lista de cursos se comprobara siempre si el curso existe y en el caso de que exista se actualizara y si no se creara el objeto

Cantidad de Cursos

```
def cantidad_cursos():
    return len(lista_objetos)
```

Retorna el tamaño de la lista de objetos que es el total de cursos aprobados

Borrar el código

```
def borrar_codigo(cod):
    for i in range (len(codis)):
        if cod==codis[i]:
            return i
```

Buscará la posición del código que se pretende borrar en la lista de códigos recorre la lista y encuentra la posición en la lista de códigos

Buscar Curso

```
def buscar_curso(cod):
    for i in range(len(lista_objetos)):
        if cod==lista_objetos[i].codigo:
            return i
```

Recibe un código y recorre la lista de objetos con el fin de encontrar la posición del curso en la lista

Agregar Curso

```
def agregar_cursos(cod,nom,pre,obl,sem,cred,es):

    if cod in codis:
        messagebox.showwarning("Alerta","el curso ya existe")
    elif cod=="" or nom=="" or obl=="" or sem=="" or cred=="" or es=="":
        messagebox.showwarning("Alerta","Rellene los campos obligatorios")
    elif int(obl)<0 or int(obl)>1:
        messagebox.showwarning("Alerta","Ingrese obligatorio [1] u Opcional [0]")
    elif int(sem)<1 or int(sem)>10:
        messagebox.showwarning("Alerta","Ingrese un semestre entre el 1 al 10")
    elif int(cred)<0 or int(cred)>10:
        messagebox.showwarning("Alerta","Ingrese un rango de creditos validos 0-10")
    elif int(es)<-1 or int(es)>1:
        messagebox.showwarning("Alerta","Ingrese aprobado [0] o cursando [1] o pendiente [-1]")
    else:
        obje=Curso(cod,nom,pre,obl,sem,cred,es)
        lista_objetos.append(obje)
        codis.append(cod)
        print(codis)
        messagebox.showinfo("Aviso","Curso Agregado Correctamente")
```

Valida los datos de la ventana agregar curso y crea un nuevo objeto del tipo curso y realiza un append a la lista de objetos

Conteo de créditos

```python
def conteo_creditos_aprobados(num_s):
    creditos_aprobados=0
    for i in range(len(lista_objetos)):
        if int(lista_objetos[i].estado)==0 and int(lista_objetos[i].semestre)==num_s:
            creditos_aprobados+=int(lista_objetos[i].creditos)
    return creditos_aprobados

def conteo_creditos_cursando(num_s_c):
    creditos_cursando=0
    for i in range(len(lista_objetos)):
        if int(lista_objetos[i].estado)==1 and int(lista_objetos[i].semestre)==num_s_c:
            creditos_cursando+=int(lista_objetos[i].creditos)
    return creditos_cursando

def conteo_creditos_pendientes(num_s_p):
    creditos_pendientes=0
    for i in range(len(lista_objetos)):
        if int(lista_objetos[i].estado)==-1 and  int(lista_objetos[i].obligatorio)==1 and int(lista_objetos[i].semestr
            creditos_pendientes+=int(lista_objetos[i].creditos)
    return creditos_pendientes

def conteo_semestre_n(N):
    conteo_sem_n=0
    for i in range(N):
        for j in range(len(lista_objetos)):
            if int(lista_objetos[j].semestre)==i+1 and int(lista_objetos[j].obligatorio)==1:
                conteo_sem_n+=int(lista_objetos[j].creditos)
    return conteo_sem_n
```


Realiza un recorrido en la lista de objetos y mediante validación de requisitos devuelve ciertos créditos según las exigencias del programa

Diagrama de Clase